REMARKS

Specification Objections

The Office Action objects Applicants' original specification for the following informalities:

- "include the 'Red Hat Package Manager' ("RPM")", citied in P.6 , Line 24, is a registered trademark
- "entries in MSDOS, and entries in a Master File Table", cited in P.13 , Line 13, is a registered trademark
- "from the executable file (204), a class to", cited in P. 12, Line 27, should be corrected as "from the executable file (304), a class to"
- "And an exec() call give its newly created process", cited in P. 13, Line 27, should be corrected as "And an exec() call gives its newly created process"

Applicants have amended the original specification in this Response in accordance with Office Action's recommendations. Applicants respectfully submit that these informalities are minor clerical errors and as such these amendments only correct the minor clerical errors, adding no new subject matter to the present application. Because these amendments correct the informalities objected to by the Office Action, the objection to Applicants' specification should be withdrawn.

Claim Rejections – 35 U.S.C. § 102 Over Brenner

Claims 1-20 stand rejected under 35 U.S.C § 102(e) as being anticipated by Brenner (U.S. Patent No. 6,859,926 B1). To anticipate claims 1-20 under 35 U.S.C. § 102(e), Brenner must disclose each and every element and limitation recited in the claims of the present application. As explained below, Brenner does not disclose each and every element and limitation recited in the claims of the present application and therefore does not anticipate the claims of the present application.

Brenner Does Not Disclose Each and Every Element Of The Claims Of The Present Application

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). As explained in more detail below, Brenner does not disclose each and every element of claim 1, and Brenner therefore cannot be said to anticipate the claims of the present application within the meaning of 35 USC § 102(e).

Independent claim 1 recites:

> 1. A method for assigning computational processes in a computer system to workload management classes, the method comprising:
>
> installing on the computer system an executable file from a software installation package, wherein the software installation package includes a specification of workload management properties for the executable file, including a definition of a workload management class;
>
> executing a process in dependence upon the executable file; and
>
> assigning the process to the workload management class.

As explained in more detail below, Brenner does not disclose each and every element of claim 1, and Brenner therefore cannot be said to anticipate the claims of the present application within the meaning of 35 U.S.C. § 102(e).

Brenner Does Not Disclose Installing On the Computer System an Executable File from a Software Installation Package, Wherein the Software Installation Package Includes a Specification of Workload Management Properties For The Executable File, Including a Definition of a Workload Management Class

The Office Action takes the position that Brennan at column 5, lines 39-46; column 6, lines 32-60 and Figures 5-6, discloses the first element of claim 1: installing on the computer system an executable file from a software installation package, wherein the software installation package includes a specification of workload management properties for the executable file, including a definition of a workload management class. Applicants respectfully note in response, however, that what Brennan at column 5, lines 39-46, in fact discloses is:

> Process Classification
>
> Each process has associated attributes which may be used to perform a classification of the process into a defined class. For example, a process may include attributes identifying the user that submitted the process, the group from which the process was submitted, the fully qualified path of the application which the process is executing, and the like. These attributes may be used with established classification rules to determine to which class the process should belong.

In addition, what Brennan at column 6, lines 32-60, in fact discloses is:

> Class Assignment Rules
>
> For a class to be defined, the class name, tier, resource shares and resource limits must be defined. Once a class has been defined, class assignment rules need to be created. The class assignment rules are used to assign processes to a class based on process attributes.
>
> FIG. 5 is an exemplary diagram illustrating classes and class assignment rules. As shown in FIG. 5, the process attributes utilized in this particular example for classification of processes are user name, group name and application path name.
>
> For example, the class assignment rule for assigning processes into the class "promoted" is that the user name be "sally", the group name be "staff", and the application path be "/bin/ksh" or "/bin/sh." Similarly, for a

> process to be classified in the "skilled" class, the group name must be "webmasters" and the application path must be "/bin/emacs." Using these class assignment rules, processes are classified into various defined classes. FIG. 6 is a diagram illustrating this procedure. As shown in FIG. 6, classification rules 660 are applied to processes 610-650. Based on the attributes of the processes, e.g., user name, group name, fully qualified path, and the like, these processes meet certain requirements of various ones of the classification rules. As a result, the processes are classified into one of the predefined classes 670 or 680. As a result of the classification, these processes now belong to classes which have an assigned tier value and number of shares which will be used to determine their access to system resources.

That is, Brennan at column 5, lines 39-46, column 6, lines 32-60 and Figures 5-6, discloses class assignment rules for assigning processes to a class based on process attributes. Neither Brennan's class assignment rules nor Brennan's process attributes disclose a specification of workload management properties for the executable file, including a definition of a workload management class as claimed in the present application. Brennan's class assignment rules for assigning processes to a class are not specified in a software installation package as claimed in the present application. In fact, Brennan does not disclose, at these reference points or anywhere else, a software installation package as claimed in the present application. A software installation package as claimed in the present application includes a specification of workload management properties for an executable file, including a definition of a workload management class. Instead of a software installation package that includes a specification of workload management properties for an executable file, Brennan only discloses that "each process has associated attributes which may be used to perform a classification of the process into a defined class." Brennan's process attributes are also not specified in a software installation package as claimed in the present application. Because neither Brennan's class assignment rules nor Brennan's process attributes are specified in a software installation package as claimed in the present application, Brennan cannot disclose installing on the computer system an executable file from such a software installation package, wherein the software installation package includes a specification of workload management properties for the executable file, including a definition of a workload management class as claimed in the present application. Because Brennan does not disclose each and every element and limitation of Applicants' claims, Brennan does not anticipate Applicants' claims, and the rejections under 35 USC § 102(e) should be withdrawn.

Relations Among Claims

Independent claims 9 and 15 are system and computer program product claims, respectively, for assigning computational processes in a computer system to workload management classes corresponding to independent method claim 1 that include a method for assigning computational processes in a computer system to workload management classes. Claim 1 is allowable for the reasons set forth above. Claims 9 and 15 are allowable for the same reasons that claim 1 is allowable. The rejections of claims 9 and 15 therefore should be withdrawn, and claims 9 and 15 should be allowed.

Claims 2-8, 10-14, and 16-20 depend respectively from independent claims 1, 9, and 15. Each dependent claim includes all of the limitations of the independent claim from which it depends. Because Brennan does not disclose or enable each and every element of the independent claims, Brennan does not disclose or enable each and every element of the dependent claims of the present application. As such, the rejections of claims 2-8, 10-14, and 16-20 should also be withdrawn, and the claims should be allowed.

Conclusion

Claims 1-20 stand rejected under 35 U.S.C. § 102 as being anticipated by Brennan. Brennan does not disclose each and every element of Applicants' claims and does not enable Applicants' claims. Brennan therefore does not anticipate Applicants' claims. Claims 1-20 are therefore patentable and should be allowed. Applicants respectfully request reconsideration of claims 1-20.

The Commissioner is hereby authorized to charge or credit Deposit Account No. 09-0447 for any fees required or overpaid.

Respectfully submitted,

Date: August 30, 2007 By: John Biggers

John Biggers
Reg. No. 44,537
Biggers & Ohanian, LLP
P.O. Box 1469
Austin, Texas 78767-1469
Tel. (512) 472-9881
Fax (512) 472-9887
ATTORNEY FOR APPLICANTS